

PROJECT / GROUP NAME	Group 14		
Start Date	2025.2.25	Finish Date	2025.3.10
Aim / Objective	Responsible for requirements research and system architecture planning		
Work package Manager	Haoxiang Han		
Contributors to this package	All members		
Description / Activities	<p>Task 1.1 Collect user requirements :</p> <ul style="list-style-type: none"> -1.1.1 Communicate with all stakeholders involved in the project to clarify their needs for the system -1.1.2 Determine the core functions, performance requirements, non-functional requirements that the system must implement <p>Task 1.2 Develop the overall project architecture design :</p> <ul style="list-style-type: none"> -1.2.1 Based on the collected requirements, design the overall architecture of the system and clarify how the various parts of the system interact -1.2.2 Identify programming languages, frameworks, databases -1.2.3 Choose the appropriate design pattern <p>Task 1.3 Design Module Interfaces</p> <ul style="list-style-type: none"> -1.3.1 Based on the architecture design, determine how the modules will communicate and exchange data. <p>Task 1.4 Complete System Requirements Document</p> <ul style="list-style-type: none"> -1.4.1 Based on the user requirements and technical architecture design collected in the earlier stages, compile them into a formal requirements document 		
Milestones		Week	
	M 1.1 Normally around week 2	2	
	M 1.2 Normally around week 2/3	2/3	
	M 1.3 Normally around week 2/3	2/3	
	M 1.4 Normally around week 3	3	
Deliverables		Week	
	D 1.1 system requirements document -A formal requirements document based on user requirements and technical architecture design	3	

PROJECT / GROUP NAME	Group 14		
Start Date	25/2/2025	Finish Date	10/6/2025
Aim / Objective	Database design and establish		
Work package Manager	Ruiyang Liu		
Contributors to this package	All members		
Description / Activities	<p>Task 2.1 Database requirement analysis</p> <ul style="list-style-type: none"> -2.1.1 Review exist documents, understand the database requirement. -2.1.2 Have a team meeting to gather detailed database requirement. -2.1.3 Determine the entity and the attribute of the database. -2.1.4 Define the quantitative relation between entities. <p>Task 2.2 Database design</p> <ul style="list-style-type: none"> -2.2.1 Draw the E-R diagram -2.2.2 Define appropriate data types, constraints (such as primary keys, foreign keys, and unique constraints) for each column in the database tables. <p>Task 2.3 Database Implementation</p> <ul style="list-style-type: none"> -2.3.1 Choose the correct database management system base on the database requirement. -2.3.2 Create the database (entity, attribute,relationship). -2.3.3 Add some data to test. <p>Task 2.4 Database testing</p> <ul style="list-style-type: none"> -2.4.1 Conduct unit testing on individual database objects -2.4.2 Perform integration testing to verify the interaction between different database components 		
Milestones		Week	
	M 2.1 & M 2.2 Normally around week 1 M 2.3 & M 2.4 Normally around week 2/3	1 2/3	
Deliverables		Week	
	D2.1 Database Requirement Specification Document D2.2 ER Diagram D2.3 SQL Scripts for Database Creation and Sample Data Insertion D2.4 Database Testing Report	D2.1 Database Requirement Specification Document D2.2 ER Diagram D2.3 SQL Scripts for Database Creation and Sample Data Insertion D2.4 Database Testing Report	

PROJECT / GROUP NAME	Group 14		
Start Date	25/2/2025	Finish Date	10/6/2025
Aim / Objective	Responsible for front-end design and user interface.		
Work package Manager	Jiawei Xing		
Contributors to this package	All members		
Description / Activities	<p>Task 3.1: Requirements Analysis and Design Planning</p> <ul style="list-style-type: none"> Gather and analyze the user requirements for the front end and user interface (UI). Research design patterns and technologies (e.g., HTML, CSS, JavaScript). Create initial wireframes and design prototypes for the application. Define the layout and navigation flow of the application. <p>Task 3.2: UI Component Development</p> <ul style="list-style-type: none"> Develop the main UI components: login page, dashboard, navigation bar, and content display areas. Implement responsive design for various screen sizes (desktop, tablet, mobile). Ensure user-friendly and intuitive navigation. Start integrating visual elements like icons, buttons, and text inputs. <p>Task 3.3: Integration with Backend Services</p> <ul style="list-style-type: none"> Work with backend developers to integrate frontend UI with API endpoints. Implement frontend validation for user input. Ensure real-time data fetching and interaction between UI components and the backend. <p>Task 3.4: Testing and Debugging</p> <ul style="list-style-type: none"> Perform unit testing for individual UI components. Conduct usability testing with the team to ensure a smooth user experience. Debug any issues related to UI rendering, layout, or interactions. Optimize the UI for performance (e.g., fast loading times, smooth animations). <p>Task 3.5: User Interface Refinements and Enhancements</p> <ul style="list-style-type: none"> Refine UI based on feedback from internal testing. Improve visual elements, animations, and transitions to enhance user experience. Ensure accessibility (color contrast, screen reader support, etc.). Implement final adjustments for the look and feel of the interface. <p>Task 3.6: Final Review and Documentation</p> <ul style="list-style-type: none"> Finalize all front-end design features. Review and document the entire front-end design and development process. Prepare for integration with the overall project and provide the final deliverables. 		
Milestones		Week	
	<ul style="list-style-type: none"> M 3.1 (Week 4): Completion of requirements gathering, wireframes, and design prototypes. M 3.2 (Week 6): First iteration of UI components (login, dashboard, navigation) completed. M 3.3 (Week 9): Integration with backend services for API interactions and real-time data fetching. M 3.4 (Week 11): UI component testing and debugging phase. 	<div>4</div> <div>6</div> <div>9</div> <div>11</div>	

	<ul style="list-style-type: none"> • M 3.5 (Week 13): UI refinements, accessibility features, and final adjustments. • M 3.6 (Week 16): Final UI deliverables and integration with the overall project. 	13 16
Deliverables		Week
	<ul style="list-style-type: none"> • D 3.1 (Week 4): Wireframes, design prototypes, and initial UI design documentation. • D 3.2 (Week 6): Completed UI components (login, dashboard, navigation) integrated with backend APIs. • D 3.3 (Week 9): Functional and tested UI with real-time data integration. • D 3.4 (Week 11): Completed UI testing and bug fixes. • D 3.5 (Week 13): Enhanced and refined UI with full functionality and accessibility features. • D 3.6 (Week 16): Final, fully integrated UI ready for project release. 	4 6 9 11 13 16

PROJECT / GROUP NAME	Group 14		
Start Date	2025.3.14	Finish Date	2025.6.10
Aim / Objective	Testing and Quality Assurance		
Work package Manager	Xiaopeng Zhang		
Contributors to this package	All		
Description / Activities	<p>Task 4.1 Functional Testing: Functional testing will ensure that each feature of the website works as expected.</p> <p>Task 4.2 Usability Testing: Usability testing will assess how user-friendly and intuitive the website is for users of different backgrounds</p> <ul style="list-style-type: none"> ▪ 4.2.1 Navigation Testing: Verify that the site's structure is easy to navigate, with clearly defined menus and categories. ▪ 4.2.2 User Interface (UI) Testing: Assess the visual appeal and user interaction with different elements, ensuring they are visually consistent with the company's branding <p>Task 4.3 Performance Testing: Performance testing will ensure that the website performs optimally under various conditions.</p> <ul style="list-style-type: none"> ▪ 4.3.1 Load Testing: Simulate high traffic to evaluate how the website handles increased user interactions. ▪ 4.3.2 Speed Testing: Check that the website loads quickly, even with high-resolution images and videos. ▪ 4.3.3 Stress Testing: Test the limits of the website's infrastructure to handle unexpected spikes in user load. <p>Task 4.4 Security Testing: Security testing will ensure that user data is protected, particularly for e-commerce transactions and personal health information.</p> <ul style="list-style-type: none"> ▪ 4.4.1 Data Encryption: Test the encryption of sensitive data, especially in payment systems and health-related records. ▪ 4.4.2 Vulnerability Scanning: Conduct tests to identify any security vulnerabilities within the website's code. ▪ 4.4.3 Access Control: Verify that user roles and permissions are correctly implemented to prevent unauthorized access. 		
Milestones		Week	
	<ul style="list-style-type: none"> - M 4.1: Initial functional testing and UI/UX review. - M 4.2: Performance testing, including load and speed tests. - M 4.3: Interim report – focus on integrated features and SDG compliance. - M 4.4: Final functional and security testing. 	4/5 6/7 9 11	
Deliverables		Week	
	D 4.1: Test cases for functional and usability testing. D 4.2: Interim report with identified bugs and resolved issues. D 4.3: Final test results, including performance and security testing outcomes.	4/5 9 13/final week	

PROJECT / GROUP NAME	Group 14 / SKT		
Start Date	2025.3.10	Finish Date	2025.5.5
Aim / Objective	Back-end development and project integration		
Work package Manager	Xuning Wei		
Contributors to this package	All members		
Description / Activities	<p>Task 5.1: Project Setup and Core Infrastructure</p> <p>Task 5.2: Product Catalog Backend 5.2.1: Implement API endpoints for retrieving product information (name, description, nutritional information, images, etc.). 5.2.2: Develop the functionality for displaying product categories and subcategories. 5.2.3: Implement search functionality for the product catalog. 5.2.4: Develop API endpoints for handling user-uploaded food information (validation, storage). Consider moderation workflows.</p> <p>Task 5.3: Nutrition Calculation and Recommendation Engine 5.3.1: Implement the nutrition calculation logic based on user input (age, height, weight, goals). 5.3.2: Design and implement algorithms for calculating required nutrient intake based on user goals (weight loss, muscle gain, general health). 5.3.3: Develop the recommendation engine to suggest suitable food items (prioritizing the company's products) based on the calculated nutrient needs. 5.3.4: API integration with product catalog to retrieve nutritional data.</p> <p>Task 5.4: User Community Features 5.4.1: Implement API endpoints for posting, retrieving, updating, and deleting user posts. 5.4.2: Implement user commenting and rating functionalities. 5.4.3: Develop forum and discussion board functionalities. 5.4.4: Implement features to moderate content and flag inappropriate posts. 5.4.5: Implement reporting mechanisms for flagging inappropriate content</p> <p>Task 5.5: E-commerce Integration and Subscription Services 5.5.1: Integrate with a payment gateway (e.g., Stripe, PayPal). 5.5.2: Implement order processing and management functionalities. 5.5.3: Develop the subscription service functionality (recurring billing, delivery scheduling). 5.5.4: Create APIs for managing user orders and subscriptions. 5.5.5: Implement security measures to protect payment information.</p> <p>Task 5.6: AI Personalization Integration and Analytics 5.6.1: Design and implement the health profile data storage and retrieval system. 5.6.2: Integrate the AI/ML algorithms for personalized food recommendations. 5.6.3: Develop the smart shopping list generation functionality. 5.6.4: Implement data analytics tracking to monitor platform usage and identify areas for improvement.</p>		
Milestones			Week
	M 5.1: Product Catalog MVP		4/5
	M 5.2: Nutrition Calculation Engine Ready		6/7
	M 5.3: Interim Report & Functional Core		9

	M 5.4: User Community Features Integrated	11
	M 5.5: E-commerce and Subscription Services MVP	13
	M 5.6: Final Release	16
Deliverables		Week
	D 5.1: Product Catalog API Functional API for browsing and searching products.	4/5
	D 5.2: Nutrition Calculation Engine Module for calculating nutrient needs and recommending products.	6/7
	D 5.3: Interim Report Progress Report	9
	D 5.4: User Community API API for posting, commenting, and forum features.	11
	D 5.5: E-commerce Integration Functional e-commerce integration with payment gateway.	13
	D 5.6: AI Personalization and Final Release Fully integrated and tested application with AI personalization.	16

PROJECT / GROUP NAME	Group 14		
Start Date	2025.6.11	Finish Date	2025.7.10
Aim / Objective	Documentation and deployment		
Work package Manager	Shao Jun Wang		
Contributors to this package	All members		
Description / Activities	<p>Task 6.1 Documentation Consolidation and Standardization</p> <ul style="list-style-type: none"> 6.1.1 Document Collection and Alignment: Aggregate deliverables from frontend, backend, database, security, and testing modules, ensuring consistency in technical terminology and version numbers. <ul style="list-style-type: none"> 6.1.1.1 Verify API documentation matches backend code interfaces (e.g., Swagger vs. actual routes). 6.1.1.2 Check consistency between database ER diagrams and SQL script table structures. 6.1.2 Format Standardization: Refactor all documentation using Markdown, integrate into a Confluence knowledge base, and generate PDF archival versions. <ul style="list-style-type: none"> 6.1.2.1 Define document templates (heading hierarchy, code block formatting, diagram labeling rules). 6.1.2.2 Automatically generate table of contents and version changelogs (e.g., via GitHub Actions). 6.1.3 Overview Documentation Authoring: Create system architecture diagrams and technology stack descriptions, highlighting module dependencies. <p>Task 6.2 Deployment Process Automation</p> <ul style="list-style-type: none"> 6.2.1 Deployment Script Development: Write Ansible Playbooks and Terraform configurations for one-click deployment across environments (dev/test/prod). <ul style="list-style-type: none"> 6.2.1.1 Automate frontend static resource deployment to CDN (e.g., AWS S3 + CloudFront). 6.2.1.2 Configure database initialization scripts (including table creation and baseline data insertion). 6.2.2 CI/CD Integration: Configure Jenkins/GitLab CI pipelines to trigger documentation generation and version releases. <ul style="list-style-type: none"> 6.2.2.1 Integrate Swagger UI for automatic API documentation updates. 6.2.2.2 Containerized deployment: Build Docker images and push to a private registry (e.g., Harbor). 6.2.3 Rollback Mechanism Design: Develop a 5-minute rollback plan using Docker image versions and database backups. <p>Task 6.3 Deliverable Review and Acceptance</p> <ul style="list-style-type: none"> 6.3.1 Internal Audit: Conduct cross-team meetings to validate consistency between documentation, code, and deployment configurations. <ul style="list-style-type: none"> 6.3.1.1 Verify API document fields match backend models (e.g., data types, mandatory fields). 		

